Data Discovery in RDF Graphs

Ioana Manolescu

INRIA and Ecole Polytechnique, France
ioana.manolescu@inria.fr
http://pages.saclay.inria.fr/Ioana.Manolescu

DEXA Conference, Aug 29, 2017

Outline

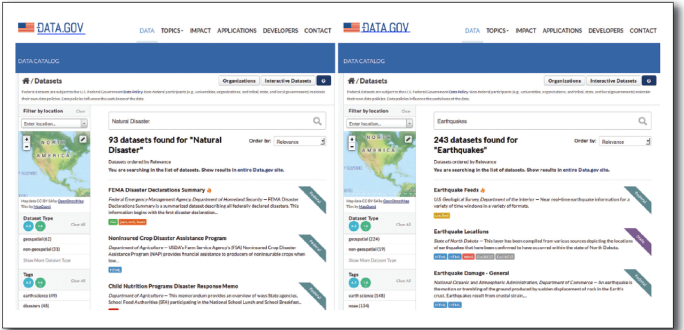
1. Background: semantic RDF graphs
2. Summarizing semantic-rich RDF graphs [ČGM15a, ČGM15b, ČGM17a]
   - Joint work with Šejla Čebirić (Inria) and François Goasdoué (U. Rennes 1 and Inria)
3. Finding insights in RDF graphs [DMS17]
   - Joint work with Yanlei Diao and Shu Shang (Ecole Polytechnique and Inria)

Part I

Background: RDF graphs

Big Data needs semantics

AI Magazine, Spring 2015



Do we really need the semantics?

Yes. All the time.

Application knowledge / constraints:

- Every Senator is an ElectedOfficial which is a Person
- (On Wikipedia) being BornInAPlace means being a Person

Do we really need the semantics?

Yes. All the time.

Application knowledge / constraints:

- Every Senator is an ElectedOfficial which is a Person
- (On Wikipedia) being BornInAPlace means being a Person

Without the semantics, we may miss query answers

Data	Constraints	Query
John is a Senator	Every Senator is a Person	Who is a Person?

Do we really need the semantics?

Yes. All the time.

Application knowledge / constraints:

- Every Senator is an ElectedOfficial which is a Person
- (On Wikipedia) being BornInAPlace means being a Person

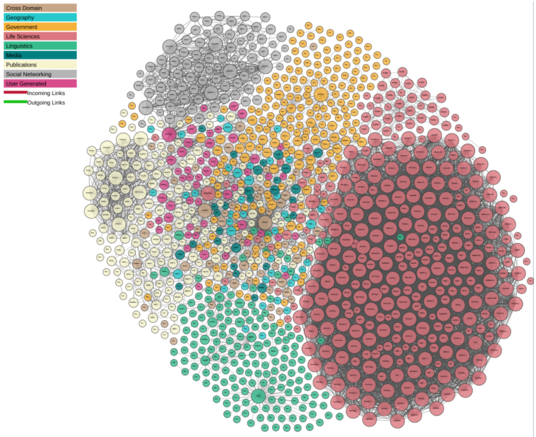
Semantic contraints are a compact way of encoding information

"Every ElectedOfficial is a Person" stated only once even if thousands of ElectedOfficials.

Semantics for Web data

Data and metadata on the Web is often structured in graphs, e.g., **RDF** (W3C's Resource Description Framework)

- Famous application: the Linked Open Data cloud (2017)



The Resource Description Framework (RDF)

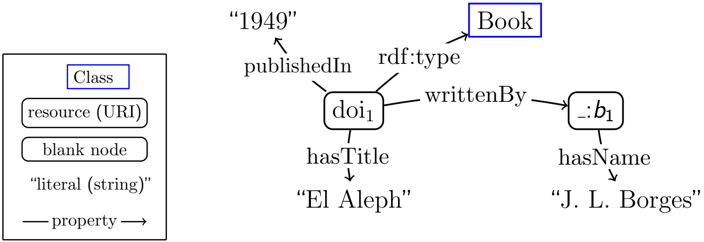
RDF graph: set of triples

Assertion	Triple	Relational notation	Intuition
Class	s rdf:type o	$o(s)$	"s is an o"
Property	s p o	$p(s, o)$	"The p of s is o"

The Resource Description Framework (RDF)

RDF graph: set of triples

Assertion	Triple	Relational notation	Intuition
Class	s rdf:type o	o(s)	"s is an o"
Property	s p o	p(s, o)	"The p of s is o"

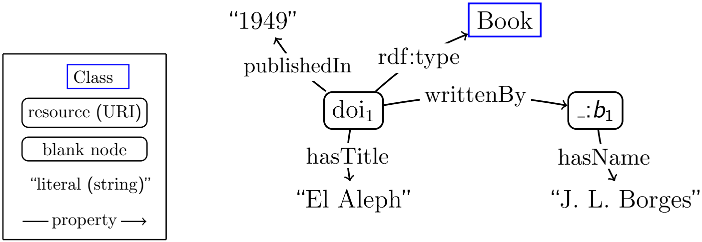


The Resource Description Framework (RDF)

Assertion	Triple	Relational notation	Intuition
Class	s rdf:type o	o(s)	"s is an o"
Property	s p o	p(s, o)	"The p of s is o"

The Resource Description Framework (RDF)

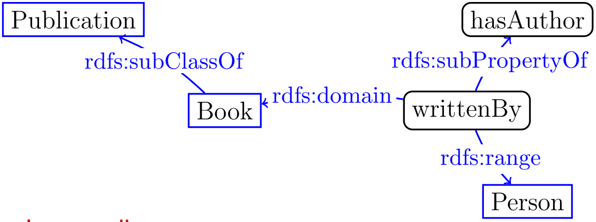
Assertion	Triple	Relational notation	Intuition
Class	s rdf:type o	o(s)	"s is an o"
Property	s p o	p(s, o)	"The p of s is o"



RDF Schema (RDFS)

Declare deductive constraints between classes and properties

Constraint	Triple	OWA interpretation
Subclass	c_1 rdfs:subClassOf c_2	$c_1 \subseteq c_2$
Subproperty	p_1 rdfs:subPropertyOf p_2	$p_1 \subseteq p_2$
Domain typing	p rdfs:domain c	$\Pi_{\text{domain}}(p) \subseteq c$
Range typing	p rdfs:range c	$\Pi_{\text{range}}(p) \subseteq c$

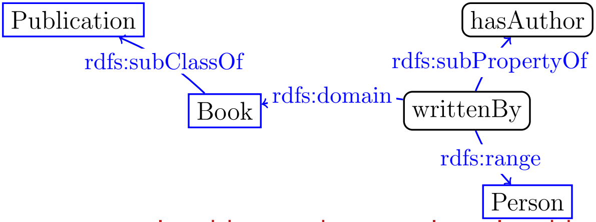


"Any c_1 is also a c_2"

RDF Schema (RDFS)

Declare deductive constraints between classes and properties

Constraint	Triple	OWA interpretation
Subclass	c_1 rdfs:subClassOf c_2	$c_1 \subseteq c_2$
Subproperty	p_1 rdfs:subPropertyOf p_2	$p_1 \subseteq p_2$
Domain typing	p rdfs:domain c	$\Pi_{\text{domain}}(p) \subseteq c$
Range typing	p rdfs:range c	$\Pi_{\text{range}}(p) \subseteq c$

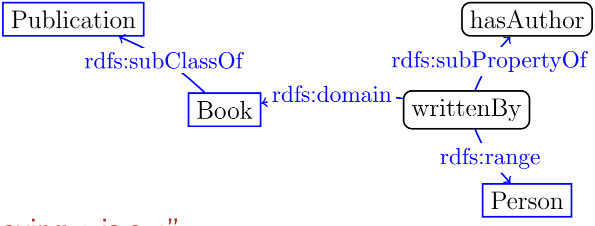


"If two resources are related by p_1, they are also related by p_2"

RDF Schema (RDFS)

Declare deductive constraints between classes and properties

Constraint	Triple	OWA interpretation
Subclass	c_1 rdfs:subClassOf c_2	$c_1 \subseteq c_2$
Subproperty	p_1 rdfs:subPropertyOf p_2	$p_1 \subseteq p_2$
Domain typing	p rdfs:domain c	$\Pi_{\text{domain}}(p) \subseteq c$
Range typing	p rdfs:range c	$\Pi_{\text{range}}(p) \subseteq c$

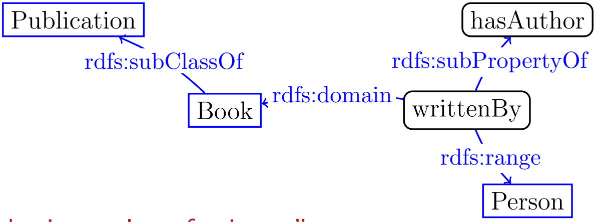


"Anyone having p is a c"

RDF Schema (RDFS)

Declare deductive constraints between classes and properties

Constraint	Triple	OWA interpretation
Subclass	c_1 rdfs:subClassOf c_2	$c_1 \subseteq c_2$
Subproperty	p_1 rdfs:subPropertyOf p_2	$p_1 \subseteq p_2$
Domain typing	p rdfs:domain c	$\Pi_{\mathrm{domain}}(p) \subseteq c$
Range typing	p rdfs:range c	$\Pi_{\mathrm{range}}(p) \subseteq c$



"Anyone who is a value of p is a c"

Open-world assumption and RDF entailment

RDF data model based on the **open-world assumption**.

Deductive constraints lead to **implicit triples**:
part of the graph even though not explicitly present

Open-world assumption and RDF entailment

RDF data model based on the **open-world assumption**.

Deductive constraints lead to **implicit triples**:
part of the graph even though not explicitly present

$$\boxed{\begin{array}{c}\textbf{explicit triples}\\ + \\ \textbf{entailment rules}\end{array} \quad \rightarrow \quad \textbf{implicit triples}}$$

Open-world assumption and RDF entailment

RDF data model based on the **open-world assumption**.

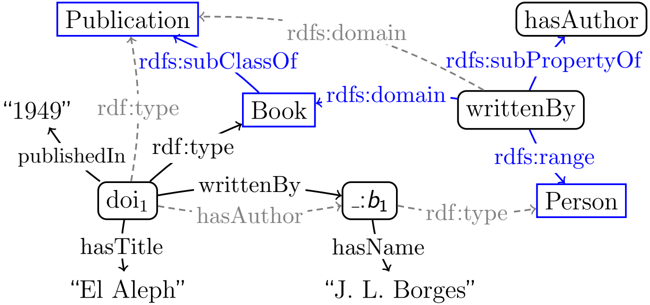
Deductive constraints lead to **implicit triples**:
part of the graph even though not explicitly present

explicit triples + **entailment rules**	$\rightarrow$	**implicit triples**

Exhaustive application of entailment leads to **saturation** (**closure**)

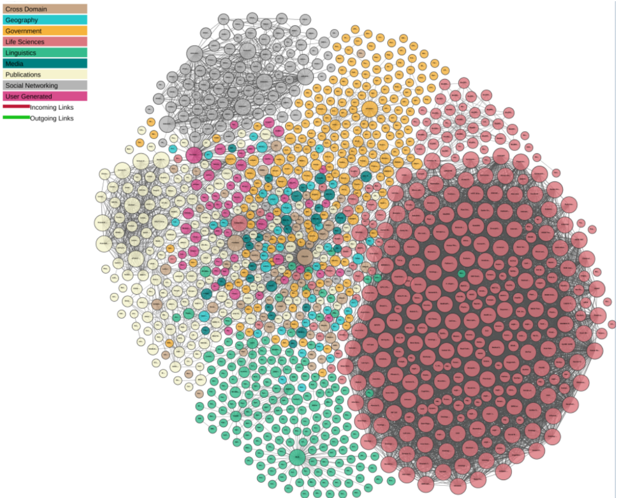
The semantics of an RDF graph $\mathtt{G}$ is its saturation $\mathtt{G}^\infty$

Sample instance entailment rules from schema and instance triples
$\mathtt{c_1}$ rdfs:subClassOf $\mathtt{c_2}$ $\wedge$ $\mathtt{s}$ rdf:type $\mathtt{c_1} \vdash_{\text{RDF}}$ $\mathtt{s}$ rdf:type $\mathtt{c_2}$
$\mathtt{p_1}$ rdfs:subPropertyOf $\mathtt{p_2}$ $\wedge$ $\mathtt{s}$ $\mathtt{p_1}$ $\mathtt{o} \vdash_{\text{RDF}}$ $\mathtt{s}$ $\mathtt{p_2}$ $\mathtt{o}$
$\mathtt{p}$ rdfs:domain $\mathtt{c}$ $\wedge$ $\mathtt{s}$ $\mathtt{p}$ $\mathtt{o} \vdash_{\text{RDF}}$ $\mathtt{s}$ rdf:type $\mathtt{c}$
$\mathtt{p}$ rdfs:range $\mathtt{c}$ $\wedge$ $\mathtt{s}$ $\mathtt{p}$ $\mathtt{o} \vdash_{\text{RDF}}$ $\mathtt{o}$ rdf:type $\mathtt{c}$



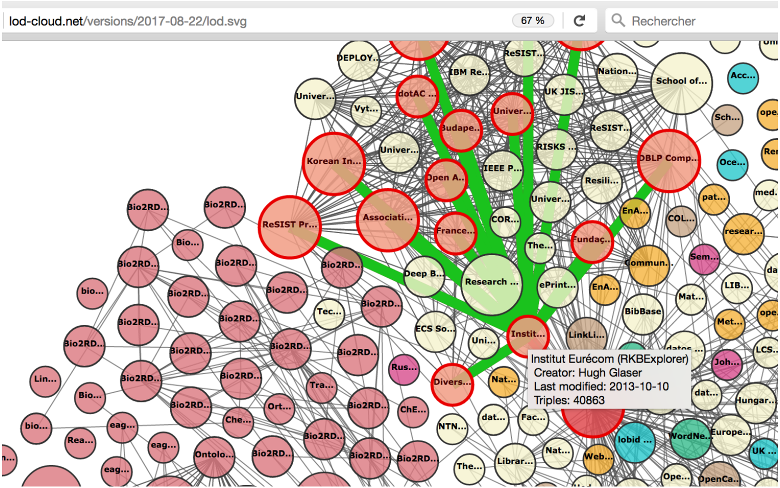
RDF graph discovery

An RDF graph can be large and complex, lack a fixed schema, include many heterogeneous values...



RDF graph discovery

An RDF graph can be large and complex, lack a fixed schema, include many heterogeneous values...



RDF graph discovery

Two approaches:

1. RDF summarization: compactly representing the explicit and implicit structure of a graph
2. Insight discovery in RDF graphs: automatically identify aggregation queries with interesting results

Part II

RDF summarization

RDF summaries

Problem

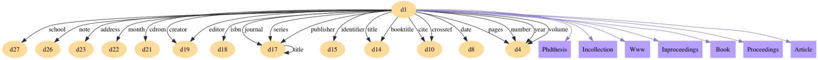
RDF graph G is large, heterogeneous, partially implicit.
How to compactly represent all its structure?

Existing solutions

Partial representation (frequent patterns, statistics etc.) e.g., [NM11, LYL13]
Potentially not compact e.g., [GW97, CFKP15]
Only for **explicit data**, e.g., [CDT13, ZDYZ14]

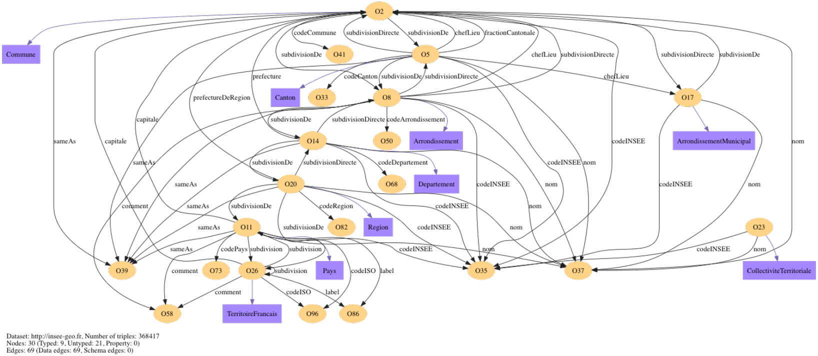
A summary of DBLP data

150M triples



A summary of geographic data

French territory division in regions, departments, urban areas, cities, districts etc.
368K triples



RDF summaries

We define

1. RDF node equivalence relation $\equiv$: equivalence relation such that class and property nodes are only equivalent to themselves
2. RDF summary $\mathtt{G}_{/\equiv}$ of an RDF graph $\mathtt{G}$: the **quotient** of $\mathtt{G}$ through $\equiv$

Recall: quotient of a directed graph G by $\equiv$

$G = (V, E)$, $\equiv$ equivalence relation on V

- $G_{/\equiv}$ nodes: one for $\equiv$ equivalence class of V
- $G_{/\equiv}$ edges: $n^1_{\equiv} \xrightarrow{a} n^2_{\equiv}$ iff $\exists n_1 \xrightarrow{a} n_2 \in \mathtt{G}$ such that n_1 represented by $n^1_{/\equiv}$, n_2 represented by $n^2_{/\equiv}$

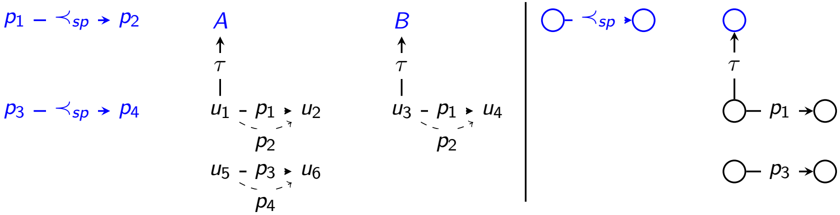
Why do we need a special RDF equivalence?

Why not use any node equivalence? E.g., forward and backward bisimilarity $\sim_{\mathtt{fb}}$ [HHK95]

Why do we need a special RDF equivalence?

Why not use any node equivalence? E.g., forward and backward bisimilarity $\sim_{\mathtt{fb}}$ [HHK95]

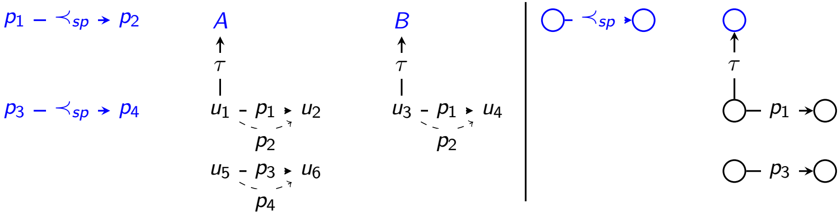
Sample graph G and its quotient through $\sim_{\mathtt{fb}}$



Why do we need a special RDF equivalence?

Why not use <u>any</u> node equivalence? E.g., forward and backward bisimilarity $\sim_{\texttt{fb}}$ [HHK95]

Sample graph $\texttt{G}$ and its quotient through $\sim_{\texttt{fb}}$

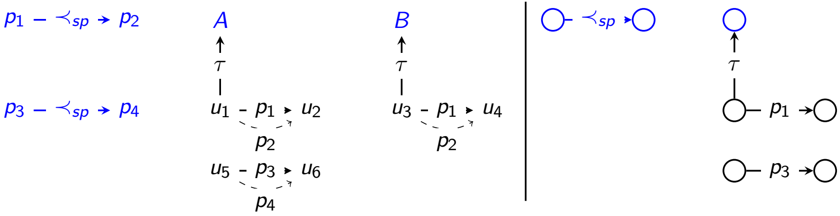


Loss of class and (some) property names

Why do we need a special RDF equivalence?

Why not use any graph node equivalence? E.g., forward and backward bisimilarity $\sim_{\mathtt{fb}}$

Sample graph $\mathtt{G}$ and its quotient through $\sim_{\mathtt{fb}}$

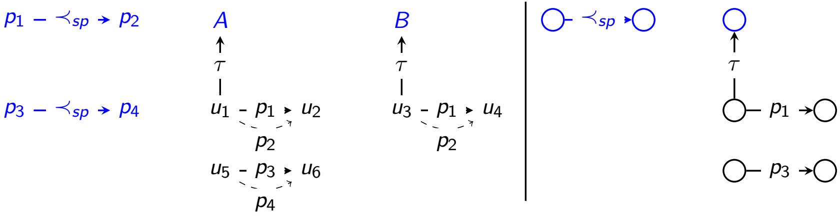


Loss of schema triples

Why do we need a special RDF equivalence?

Why not use any graph node equivalence? E.g., forward and backward bisimilarity $\sim_{\texttt{fb}}$

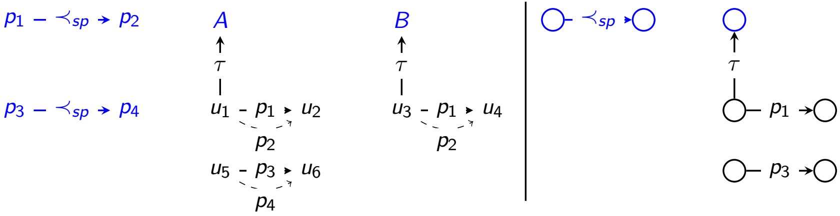
Sample graph G and its quotient through $\sim_{\texttt{fb}}$



Why do we need a special RDF equivalence?

Why not use any graph node equivalence? E.g., forward and backward bisimilarity $\sim_{\texttt{fb}}$

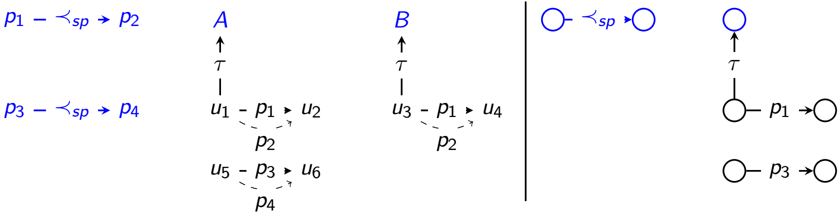
Sample graph $\texttt{G}$ and its quotient through $\sim_{\texttt{fb}}$



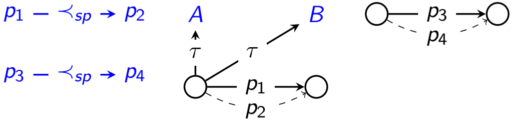
Loss of implicit triples

Why do we need a special RDF equivalence?

Sample graph G and its quotient through $\sim_{\texttt{fb}}$



Quotient of the same graph through the RDF node equivalence $\equiv_{\texttt{fb}}$



Formal summary properties

For any RDF equivalence relation $\equiv$:

Size limit	The summary is at most as large as the graph.
Schema preservation	The schema of $\mathtt{G}_{/\equiv}$ is the schema of $\mathtt{G}$.
Representativeness	Any conjunctive query q with answers on $\mathtt{G}$ also has answers on its summary: $q(\mathtt{G}^{\infty}) \neq \emptyset \;\Rightarrow\; q((\mathtt{G}_{/\equiv})^{\infty}) \neq \emptyset$ This enables **query pruning (for query answering) without saturating** $\mathtt{G}$

Which equivalence relations to use?

Equivalence notions previously studied

- Forward / backward / forward and backward simulation
- Forward / backward / forward and backward bisimulation

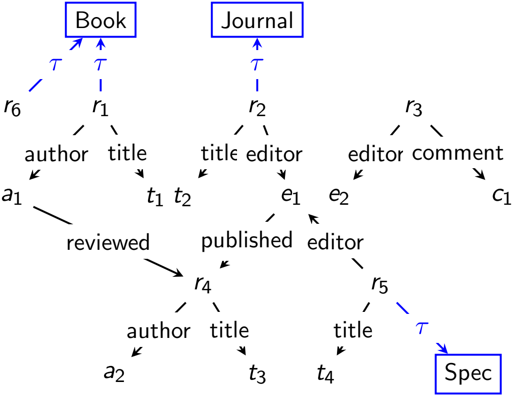
Adapted to semantic RDF graphs

Novel equivalence notions we introduce (see next)

- Flexible similarity suited to heterogeneous graphs
- Based on property cliques and possibly on RDF types

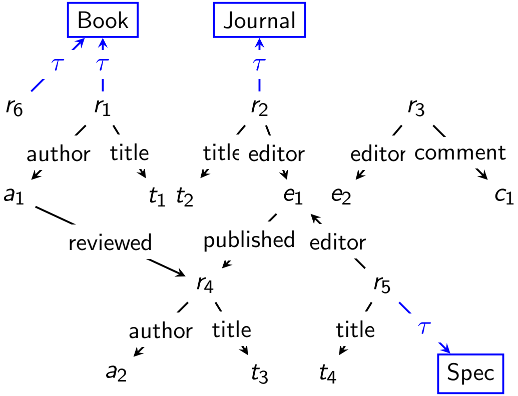
RDF node equivalence based on property cliques

Intuition: a_1, a_2 are similar; r_1, r_2, r_3, r_4, r_5 are similar



RDF node equivalence based on property cliques

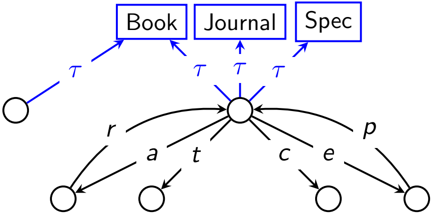
Output property cliques: $\{a, t, e, c\}$; $\{r\}$; $\{p\}$; $\emptyset$
Input property cliques: $\{a\}$; $\{t\}$; $\{e\}$; $\{c\}$; $\{r, p\}$; $\emptyset$



Weak clique-based summaries

Two nodes are weakly equivalent ($\equiv_{\mathrm{W}}$) iff they have **the same input clique** **or** **the same output clique.**

Weak summary $\mathtt{G}_{/\equiv_{\mathrm{W}}}$ of the sample RDF graph $\mathtt{G}$:

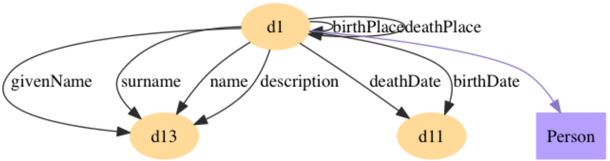


Property: In $\mathtt{G}_{/\equiv_{\mathrm{W}}}$, each data property appears exactly once ⇒ its nodes are "source of p, target of p" for each p [ČGM15b].

Weak clique-based summaries

Property: $\mathtt{G}_{/\equiv_{\mathrm{W}}}$ nodes are "source of $\mathtt{p}$, target of $\mathtt{p}$" for each $\mathtt{p}$.

Detecting errors in the data:: why do the $\mathrm{birthplace}$ and $\mathrm{deathplace}$ loop?



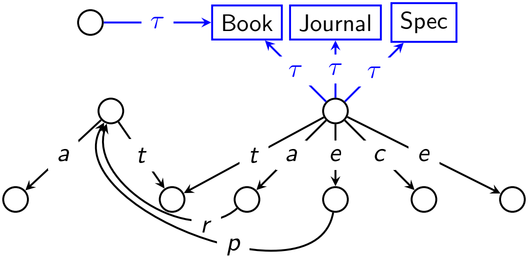
Looking in the data, we find:

⟨http://dbpedia.org/resource/Kunitomo_Ikkansai⟩ ⟨http://www.w3.org/1999/02/22-rdf-syntax-ns#type⟩ ⟨http://xmlns.com/foaf/0.1/Person⟩ .
⟨http://dbpedia.org/resource/Kunitomo_Ikkansai⟩⟨http://dbpedia.org/ontology/birthPlace⟩ ⟨http://dbpedia.org/resource/Kunitomo_Ikkansai⟩ .

Strong clique-based summaries

Two nodes are strongly equivalent ($\equiv_{\mathrm{S}}$) iff they have **the same input clique** **and** **the same output clique.**

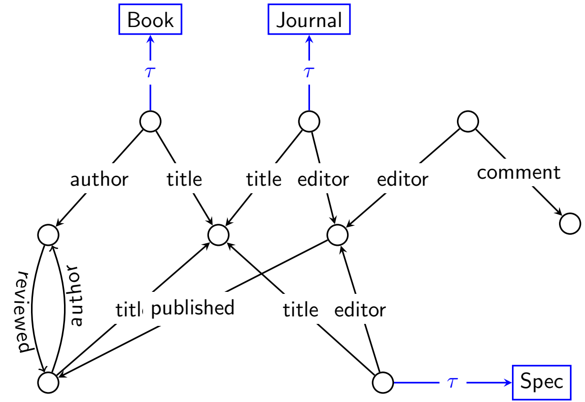
Strong summary $\mathtt{G}_{/\equiv_{\mathrm{S}}}$ of the sample RDF graph $\mathtt{G}$:



Using types for summarization

Group nodes **first by their types**; then group untyped nodes by their property cliques.

Typed weak summary $\mathtt{G}_{/\equiv_{\mathrm{TW}}}$ of the sample RDF graph $\mathtt{G}$:



On this example, this is also the typed strong summary $\mathtt{G}_{/\equiv_{\mathrm{TS}}}$.

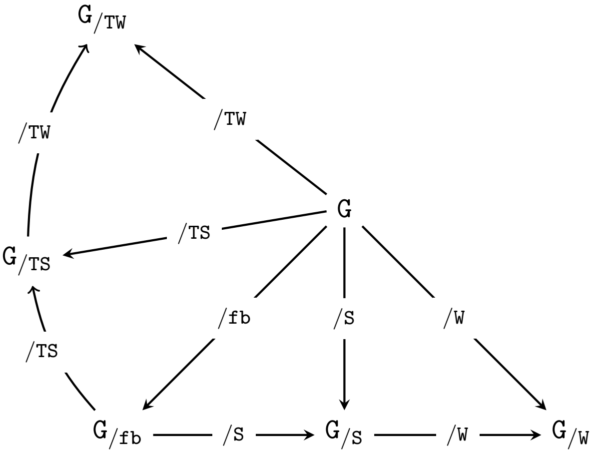
RDF summaries outline

Summary	Weak?	Strong?	Types first?
$\mathtt{G}_{/\equiv \mathrm{W}}$	✓		
$\mathtt{G}_{/\equiv \mathrm{S}}$		✓	
$\mathtt{G}_{/\equiv \mathrm{TW}}$	✓		✓
$\mathtt{G}_{/\equiv \mathrm{TS}}$		✓	✓

RDF summaries outline

Summary	Weak?	Strong?	FW bisim?	BW bisim?	Types first?
$\mathtt{G}_{/\equiv\mathrm{W}}$	✓				
$\mathtt{G}_{/\equiv\mathrm{S}}$		✓			
$\mathtt{G}_{/\equiv\mathrm{TW}}$	✓				✓
$\mathtt{G}_{/\equiv\mathrm{TS}}$		✓			✓
$\mathtt{G}_{/\equiv\mathrm{fw}}$			✓		
$\mathtt{G}_{/\equiv\mathrm{bw}}$				✓	
$\mathtt{G}_{/\equiv\mathrm{fb}}$			✓	✓	
$\mathtt{G}_{/\equiv\mathrm{fw,T}}$			✓		✓
$\mathtt{G}_{/\equiv\mathrm{bw,T}}$				✓	✓
$\mathtt{G}_{/\equiv\mathrm{fb,T}}$			✓	✓	✓

Relations between RDF summaries [ČGM17b]



Summary size comparison (more in [ČGM17b])

Graph $\mathtt{G}$	$\lvert\mathtt{G}\rvert$	Summary $\mathtt{G}_{/\equiv}$	$\lvert\mathtt{G}_{/\equiv}\rvert$	$cf_{\equiv}$
DBLP	150,787,464	$\mathtt{G}_{/\mathrm{W}}$	**71**	2,123,767
DBLP	150,787,464	$\mathtt{G}_{/\mathrm{S}}$	**206**	731,978
DBLP	150,787,464	$\mathtt{G}_{/\mathrm{fw}}$	**262,695**	574
LUBM1M	1,227,868	$\mathtt{G}_{/\mathrm{W}}$	**161**	7,579
LUBM1M	1,227,868	$\mathtt{G}_{/\mathrm{S}}$	**207**	5,903
LUBM1M	1,227,868	$\mathtt{G}_{/\mathrm{fw}}$	**1982**	617
LUBM10M	11,990,183	$\mathtt{G}_{/\mathrm{W}}$	**162**	74,013
LUBM10M	11,990,183	$\mathtt{G}_{/\mathrm{S}}$	**206**	58,204
LUBM10M	11,990,183	$\mathtt{G}_{/\mathrm{fw}}$	**24,958**	480
LUBM10M	11,990,183	$\mathtt{G}_{/\mathrm{bw}}$	**6,162**	1,944
LUBM10M	11,990,183	$\mathtt{G}_{/\mathrm{fb}}$	**11,990,076**	1

Summarizing $\mathtt{G}^\infty$

Recall: With an RDF Schema, the semantics of $\mathtt{G}$ is $\mathtt{G}^\infty \Rightarrow$
We really need $(\mathtt{G}^\infty)_{/\equiv}$!

1. Saturate $\mathtt{G}$, then summarize
2. Can we avoid saturating $\mathtt{G}$?...

Summarizing $\mathtt{G}^\infty$

Recall: With an RDF Schema, the semantics of $\mathtt{G}$ is $\mathtt{G}^\infty$ $\Rightarrow$
We really need $(\mathtt{G}^\infty)_{/\equiv}$!

1. Saturate $\mathtt{G}$, then summarize
2. Can we avoid saturating $\mathtt{G}$?...

Shortcut theorem [ČGM17a]

For the summaries $\mathtt{G}_{/\mathrm{W}}$, $\mathtt{G}_{/\mathrm{S}}$, $\mathtt{G}_{/\mathrm{fw}}$, $\mathtt{G}_{/\mathrm{bw}}$, $\mathtt{G}_{/\mathrm{fb}}$:

$$(\mathtt{G}^\infty)_{/\equiv} \text{ is the same as } ((\mathtt{G}_{/\equiv})^\infty)_{/\equiv}$$

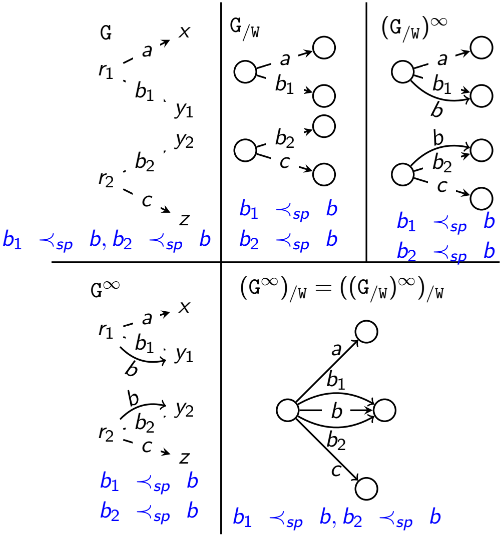
Also: **sufficient condition** for any $\equiv$ to admit the shortcut.

Shortcut toward the summary of $\mathtt{G}^{\infty}$

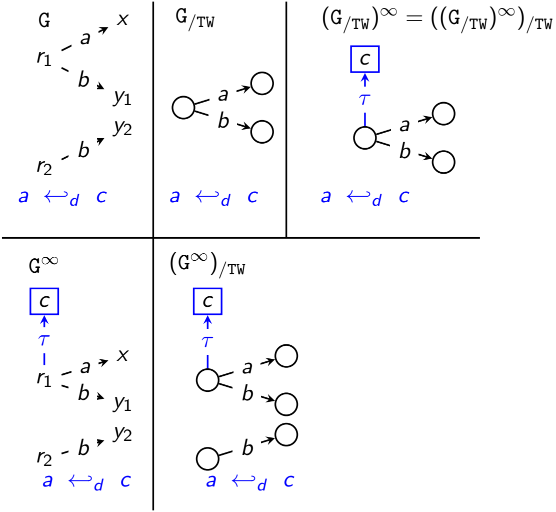
Direct	$\mathtt{G} \rightarrow$ **sat.** $\rightarrow \mathtt{G}^{\infty} \rightarrow$ **summ.** $\rightarrow (\mathtt{G}^{\infty})_{\equiv}$
Shortcut	$\mathtt{G} \rightarrow$ **summ.** $\rightarrow \mathtt{G}_{\equiv} \rightarrow$ sat. $\rightarrow (\mathtt{G}_{\equiv})^{\infty} \rightarrow$ summ. $\rightarrow ((\mathtt{G}_{\equiv})^{\infty})_{\equiv}$

If $\mathtt{G}_{\equiv}$ is much smaller than $\mathtt{G}$, **the shortcut may be faster!**
Up to 20 times in our experiments [ČGM17b]

Shortcut example: $\mathtt{G}_{\mathtt{W}}$



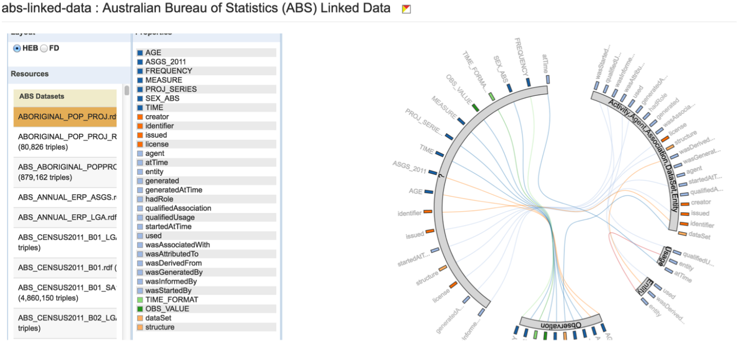
Shortcut counter-example: $\mathtt{G}_{\mathtt{TW}}$



Summary-enabled LOD cloud exploration

ILDA Inria team (E. Pietriga, H. Ozaygen)

Use summary to derive visualisation instead of the original graph (smaller, faster)

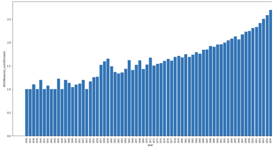


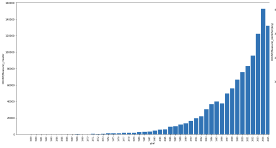
Part III

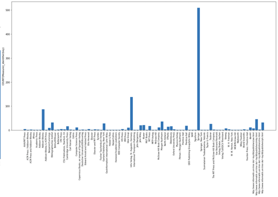
Finding insights in RDF graphs

We consider an insight to be the result of an aggregation query over the RDF graph

We focus one-dimensional aggregates ⇒ 2D layout







An insight is interesting if a certain measure (e.g., variance) on its set of aggregation values is high

Problem

Problem: given a graph G, find the top-k insights

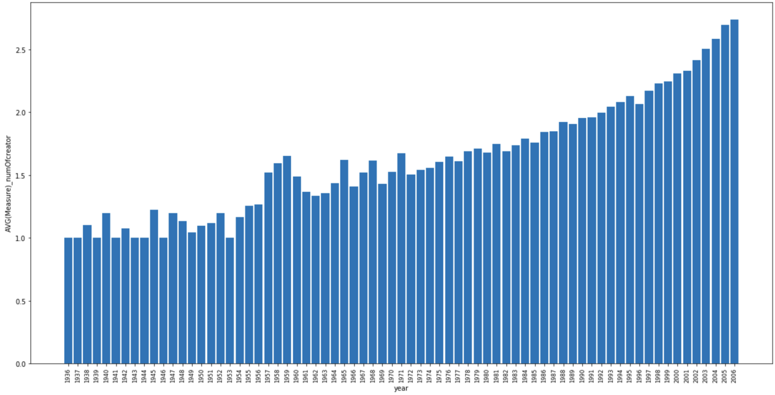
Dagger approach

Dagger: Digging for Interesting Aggregates in RDF Graphs [DMS17] (ongoing)

1. Candidate facts Resources from `G`: of a certain type, or having certain property sets
2. Candidate dimension Properties of the candidate facts, with strong support and relatively few distinct values.
Also: derived properties, e.g., authors count;
3. Candidate measure Another property of the candidate facts
Also: automatic value typing
4. Candidate aggregation function Chosen depending on the measure type

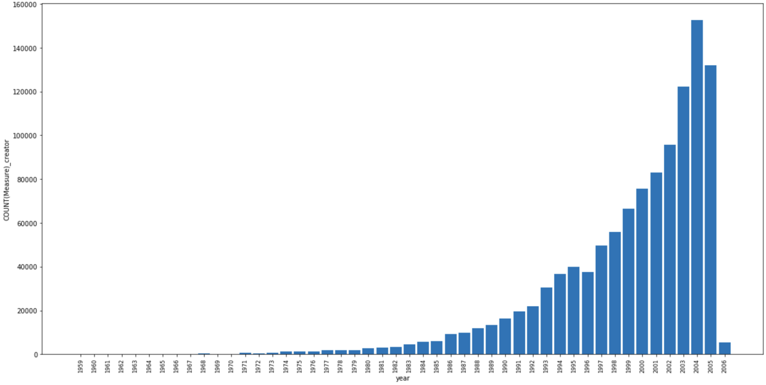
Dagger-selected aggregate in DBLP data

Average number of authors of journal articles, per publication year



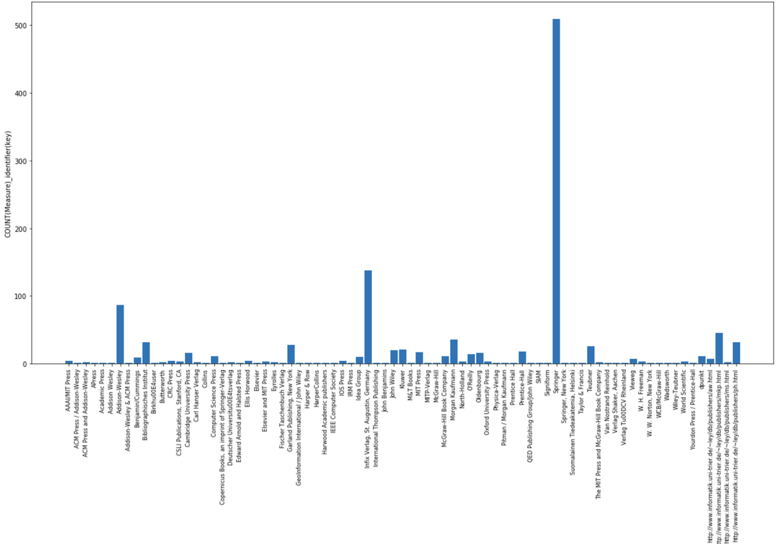
Dagger-selected aggregate in DBLP data

Number of book authors, per book publication year



Dagger-selected aggregate in DBLP data

The number of books by each publisher (highest: Springer)



Part IV

Conclusion

The need for RDF graph discovery tools

- RDF graphs can be **large and complex**, they lack a prescriptive schema
- Semantic rules lead to **implicit data**
- Toward helping users to discover RDF graphs:
  1. **Structural quotient summaries** representing the complete graph structure; compact clique-based summaries; available at:

     https://team.inria.fr/cedar/projects/rdfsummary/

  2. **Insight discovery**: interesting aggregate queries; project Web page:

     https://team.inria.fr/cedar/projects/dagger/

- Many follow-up directions: parallelization, more interestingness measures, extensions to ML.

References

[CDT13] Stéphane Campinas, Renaud Delbru, and Giovanni Tummarello. Efficiency and precision trade-offs in graph summary algorithms. In IDEAS, 2013.

[CFKP15] Mariano P. Consens, Valeria Fionda, Shahan Khatchadourian, and Giuseppe Pirrò. S+EPPs: Construct and explore bisimulation summaries + optimize navigational queries (demo). PVLDB, 8(12), 2015.

[ČGM15a] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. Query-oriented summarization of RDF graphs. In BICOD, 2015.

[ČGM15b] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. Query-oriented summarization of RDF graphs (demonstration). PVLDB, 8(12), 2015.

[ČGM17a] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. A framework for efficient representative summarization of RDF graphs. In International Semantic Web Conference (ISWC), 2017.

[ČGM17b] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. Query-Oriented Summarization of RDF Graphs. Research Report RR-8920, INRIA, 2017.

[DMS17] Yanlei Diao, Ioana Manolescu, and Shu Shang. Dagger: Digging for interesting aggregates in RDF graphs. In International Semantic Web Conference (ISWC), 2017.

References (cont.)

[GW97] Roy Goldman and Jennifer Widom. Dataguides: Enabling query formulation and optimization in semistructured databases. In VLDB, 1997.

[HHK95] Monika Rauch Henzinger, Thomas A. Henzinger, and Peter W. Kopke. Computing simulations on finite and infinite graphs. In FOCS, 1995.

[LYL13] Shou-De Lin, Mi-Yen Yeh, and Cheng-Te Li. Sampling and summarization for social networks (tutorial), 2013.

[NM11] Thomas Neumann and Guido Moerkotte. Characteristic sets: Accurate cardinality estimation for RDF queries with multiple joins. In ICDE, 2011.

[ZDYZ14] Haiwei Zhang, Yuanyuan Duan, Xiaojie Yuan, and Ying Zhang. ASSG: adaptive structural summary for RDF graph data. In ISWC (Posters and Demonstrations), 2014.